

Studi Komparasi Beberapa Pengendali untuk Pengontrolan Lampu Lalu Lintas Dua Persimpangan Berdekatan

Comparative Study of Multiple Controllers for Controlling Traffic Lights of Two Adjacent Intersections

Muhammad Yudha Permana, Muhammad Rizal Taufik, Fuzi Fauzia Ahmad, Muhammad Fahmi Ibrahim

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer

Universitas Komputer Indonesia

Email : yudhapermana@email.unikom.ac.id

Abstrak - Tujuan dilakukannya penelitian ini untuk merancang sebuah sistem lampu lalu lintas yang memiliki pewaktuan pergantian warna lampu yang efektif, merancang sebuah sistem yang dapat mengatur lampu lalu lintas menggunakan *system counter*, *adaptive fix timer*, *semi intellegent controller*, dan *vehicle actuated controller* untuk menghitung kepadatan kendaraan pada suatu jalur dalam dua persimpangan yang berdekatan agar tidak terjadi penumpukan kendaraan pada jalur tersebut, dan membandingkan semua sistem yang dibuat agar dapat di ketahui mana sistem yang lebih unggul. Sistem ini merupakan sebuah instruksi pengontrolan PLC (*Programmable Logic Controller*) yang di buat menggunakan aplikasi CX – programmer untuk mengontrol lampu lalu lintas pada simulator dua persimpangan yang dihubungkan secara serial dengan mikrokontroler Atmega2560 yang terdapat pada *hardware interface*. Metode yang dilakukan pada penelitian ini dengan mengambil data rata – rata waktu tunggu dari waktu pagi, siang dan sore hari. Sistem ini memiliki keunggulan masing – masing dalam kondisi tertentu. Hasil penelitian yang menggunakan metode tersebut didapatkan bahwa sistem *counter* memiliki total rata – rata waktu tunggu sebesar 8,04164 s, sistem *adaptive fix timer* memiliki total rata – rata waktu tunggu sebesar 10,82615 s, sistem *semi intellegent controller* memiliki total rata – rata waktu tunggu 9,99704 s, dan sistem *vehicle actuated controller* memiliki total rata – rata waktu tunggu sebesar 11,34686 s. Berdasarkan perancangan tersebut dapat disimpulkan bahwa perbandingan sistem ini berhasil dilakukan dengan sistem *counter* dikategorikan paling efektif pada kasus ini.

Kata kunci : PLC, Mikrokontroler, Hardware Interface, Labview, Serial.

Abstract - The purpose of this research is to design a traffic light system that has an effective color change timing, design a system that can set traffic lights using a Counter system, adaptive fix timer, semi intelligent controller, and vehicle actuated controller to calculate vehicle density at a lane in two adjacent intersections so that there is no buildup of vehicles on the lane, and compare all the systems made so that it can be seen which system is superior. This system is a PLC (*Programmable Logic Controller*) instruction that is made using the CX - programmer application to control the traffic lights on the two crossing simulators connected serially with the Atmega2560 microcontroller found in the hardware interface. The method used in this study is to take the average waiting time data from morning, afternoon and evening. This system has their respective advantages under certain conditions. The results of the study using this method found that the counter system has a total average waiting time of 8.04164 s, the adaptive fix timer system has a total average waiting time of 10.82615 s, the semi intelligent controller system has a total average time waiting 9.99704 s, and the vehicle actuated controller system has a total average waiting time of 11.334686 s. Based on the design, it was concluded that the comparison of this system was successfully carried out with a counter system categorized as the most effective in this case.

Keywords : PLC, Microcontroller, Hardware Interface, Labview, Serial.

I. PENDAHULUAN

A. Latar Belakang

Kemacetan yang terjadi pada persimpangan jalan yang diakibatkan oleh sistem pewaktuan lampu lalu lintas, yang di akibatkan oleh kurangnya pembagian jatah lampu hijau pada persimpangan, pada sebagian besar sistem lampu lalu lintas saat ini yaitu lampu hijau pada persimpangan disamakan di setiap jalur tanpa ada pertimbangan kepadatan kendaraan yang ada di setiap jalurnya, pada kenyataannya terdapat perbedaan kepadatan di setiap jalur lalu lintas pada persimpangan, hal ini mengakibatkan jalur yang kosong atau tidak mengalami kepadatan mendapat kesempatan lampu hijau yang lama sehingga tidak efektif, apa lagi jika ada kasus seperti ada dua persimpangan yang sangat berdekatan kepadatan pada suatu jalur sering di alami. Dengan terjadinya kepadatan pada suatu jalur yang di akibatkan oleh lampu lalu lintas yang tidak efektif maka di perlukan suatu sistem yang dapat mendeteksi kepadatan kendaraan pada suatu jalur persimpangan yang bekerja secara otomatis agar dapat terhindar dari kemacetan yang di akibatkan oleh kepadatan lalu lintas dan lampu lalu lintas yang tidak efektif.

Aria dan Faizal (2017) mengatakan bahwa semua sistem kontrol pada pengaturan lampu lalu lintas di setiap persimpangan yang ada di kota – kota besar di Negara Indonesia, sebenarnya sudah memiliki sistem kontrol yang sesuai standar dan cukup baik, karena penentuan lamanya antara lampu merah, kuning dan hijau sudah terprogram secara konstan. Namun jika dibandingkan dengan negara lain, teknologi lampu lalu lintas yang dipakai saat ini tergolong kedalam teknologi yang sudah tertinggal dikarenakan lampu lalu lintas saat ini dapat menentukan penyalan lampu lalu lintas untuk setiap ruas jalan secara adil dan cerdas. Sistem tersebut dapat secara otomatis mendeteksi kepadatan lalu lintas untuk menentukan mana saja ruas yang harus didahulukan agar lalu lintas dapat berjalan secara tertib.[1]

B. Tinjauan Start Of Art

Pada bagian tinjauan pustaka ini kita membandingkan sistem yang kita buat dengan sistem yang pernah di buat sebelumnya, di bawah ini merupakan sistem yang telah dibuat sebelumnya dan perbandingan dengan sistem yang kita buat. “Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection” [2]. Sistem ini dibuat

menggunakan *fuzzy logic*, sistem ini dapat mendeteksi kepadatan deteksi dilakukan dengan image segmentation yang diproses menggunakan filtering object dan ekstraksi blob, perbandingan sistem ini dengan sistem yang saya buat adalah sistem ini belum bisa mengendalikan atau mengatur pergantian lampu lalu lintas dengan kepadatan yang telah di deteksi. “Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16” [3]. Sistem ini dibuat menggunakan mikrokontroler AVR ATmega16 dan di program menggunakan Code Vision AVR berbasis bahasa C, perbandingan sistem ini dengan sistem yang sekarang di buat adalah sistem ini hanya menggunakan sistem pewaktuan *adaktif fiks timer* sehingga jika terdeteksi kepadatan pada suatu lintasan harus di lakukan pengatutan pewaktuan lampu lalu lintas secara manual.

“Simulasi Penerapan Anfis pada Sistem Lampu Lalu Lintas Enam Ruas” [4]. Sistem ini dibuat dengan *neuro-fuzzy* menggunakan ANFIS di persimpangan enam ruas, perbandingan sistem ini dengan sistem yang di buat sekarang adalah sistem ini belum bisa mendeteksi kepadatan pada suatu jalur dan belum bisa mengatur perubahan pewaktuan lampu lalu lintas berdasarkan kepadatan kendaraan pada lalu lintas. “*Traffic light mapping and detection*”[5]. Sistem ini dibuat dengan beberapa sensor yaitu seperti sonar, radar, dan lidar, lalu memanfaatkan dari teknologi maps untuk mendeteksi keberadaan lampu lalu lintas, dan mendeteksi keadaan lampu lalu lintas dengan kamera di atas mobil sehingga kita dapat merasakan secara visual lampu lalu lintas tersebut. Perbandingan sistem ini dengan sistem yang saya buat adalah sistem ini belum bisa mengatur lampu lalu lintas dengan menggunakan kepadatan pada suatu jalur sebagai patokannya. “*Traffic light detection for colorblind individuals*”[6]. Sistem ini dibuat menggunakan labview untuk mengolah data gambar agar membantu pengemudi yang buta warna dalam mengidentifikasi warna lampu lalu lintas. Perbedaan nya dengan sistem yang saya buat sistem ini belum bisa mengatur pewaktuan yang efektif agar tidak terjadi kemacetan pada suatu jalur tertentu dengan kepadatan kendaraan sebagai acuannya.

“*Real-time detection and classification of traffic light signals*”[7]. Sistem ini dibuat bertujuan untuk membantu para pengemudi untuk melihat lampu lalu lintas yang mungkin susah untuk di lihat agar bisa terlihat dan agar pengemudi bisa mengambil sikap lebih cermat

ketika lampu lalu lintas sudah terdeteksi. Perbedaan dengan sistem yang saya buat adalah sistem ini belum bisa menghitung kendaraan pada suatu jalur dan belum bisa mengatur perpindahan warna lampu lalu lintas di tentukan dari jumlah keoadatan oada suatu jalur. “*Robust and real-time traffic light recognition based on hierarchical vision architecture*”[8]. Sistem ini berfungsi untuk membantu pengguna mobil pintar agar dapat mengenal lampu lalu lintas yang baik dan secara *realtime* dan agar mobil tersebut dapat memutuskan arah yang mau di ambil. Perbedaan dari sistem yang saya buat adalah sistem tersebut hanya bisa mendeteksi keberadaan lampu lalu lintas belum bisa mengatur lampu lalu lintas dari kepadatan jalur. “*Priority level mutualism for emergency vehicle using game theory*”[9]. Sistem ini bisa mengatur lampu lalu lintas dari pendeteksian kendaraan darurat seperti ambulans, mobil pemadam kebakaran, mobil polisi dan lainnya sebagai acuan untuk sistem agar suatu jalur bisa di prioritaskan, perbedaan dari sistem yang saya buat adalah sistem tersebut belum bisa mengatur perubahan lampu lalu lintas dari kepadatan kendaraan dan belum bisa mendeteksi seberapa banyak kendaraan yang ada pada jalur. “*Intelligent Traffic Lights Based on RFID*”[10], Alat ini dirancang dengan RFID untuk memprioritaskan penyalan lampu lalu lintas untuk memberikan kualitas layanan untuk kendaraan darurat (ambulace) dan meningkatkan akurasi lampu lalu lintas secara otomatis.

“*A Unified Framework for Vehicle Rerouting and Traffic Light Control To Reduce Traffic Congestion*”[11]. Alat ini adalah pengembangan sistem peringatan yang terdiri dari Intelligent Traffic Lights (ITLs) yang memberikan informasi kepada pengemudi tentang kepadatan lalu lintas dan kondisi cuaca di jalan-jalan kota diusulkan dan dievaluasi melalui simulasi. “*Suspended Traffic Lights Detection and Distance Estimation Using Color Features*”. Alat ini dirancang untuk mendeteksi setiap perpindahan warna lampu pada lampu lalu lintas. “Simulasi Penerapan Anfis pada Sistem Lampu Lalu Lintas Enam Ruas” Sistem ini di buat dengan neuro-fuzzy menggunakan ANFIS di persimpangan enam ruas, perbandingan sistem ini dengan sistem yang di buat sekarang adalah sistem ini belum bisa mendeteksi kepadatan pada suatu jalur dan belum bisa mengatur perubahan pewaktuan lampu lalu lintas berdasarkan kepadatan kendaraan pada lalu lintas.

C. Tujuan

Ada beberapa tujuan yang ingin di capai dari penulisan penelitian ini yaitu merancang sebuah sistem lampu lalu lintas yang memiliki pewaktuan pergantian warna lampu yang efektif, merancang sebuah sistem *counter*, *adaptive fix timer*, *semi intelligent controller*, dan *vehicle actuated controller* untuk menghitung kepadatan kendaraan pada suatu jalur dalam dua persimpangan yang berdekatan agar tidak terjadi penumpukan kendaraan pada jalur tersebut, merancang sebuah simulator persimpangan, serta melakukan perbandingan dengan semua sistem.

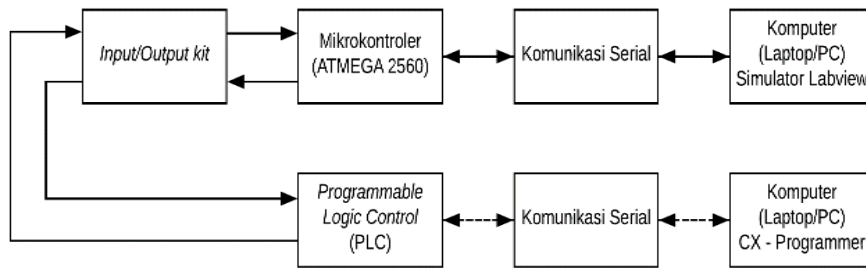
II. METODOLOGI

Ada beberapa jenis model persimpangan salah satunya persimpangan pada highway, persimpangan dalam kota, dua persimpangan berdekatan, dan lain - lain. Pada sistem ini model persimpangan yang digunakan yaitu dua persimpangan berdekatan, yang mana pada dua persimpangan berdekatan ini rentan sekali terjadi kemacetan yang diakibatkan perhitungan waktu yang tidak tepat dan kurang efektif.

Pada sistem ini akan ada beberapa sensor yang dapat membaca setiap kendaraan yang melewatinya. Sensor ini akan di fungsikan untuk mendeteksi kendaraan yang terjadi pada dua persimpangan bedekatan. Tetapi sistem pengontrolan *adaptive fix timer* tidak menggunakan sensor, karena sistem pengontrolan ini menggunakan murni fungsi timer untuk menentukan pewaktuan.

A. Perancangan Sistem

Pada perancangan sistem ini menggunakan dua program yang digunakan yaitu pertama menggunakan CX – *Programmer* yang berfungsi sebagai kontrol terhadap semua kondisi lampu lalu lintas yang terdapat pada simulator LabVIEW. Kedua menggunakan LabVIEW yang berfungsi sebagai simulator tampilan lampu lalu lintas yang terhubung secara serial dengan CX – *Programmer*. Pada sistem ini memiliki dua blok diagram sistem, pertama blok diagram sistem untuk untuk lampu lalu lintas dan yang kedua blok diagram sistem pendeteksian kendaraan. Berikut blok diagram dapat dilihat pada **Gambar 1**.



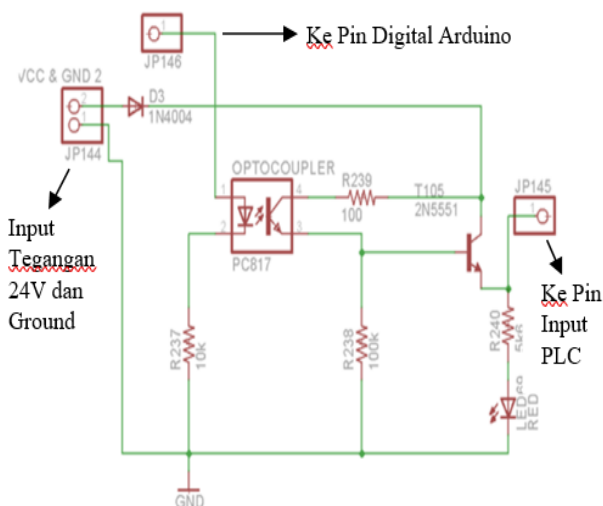
Gambar 1. Blok Diagram Hardware Interface PLC dan PC Pada Pengontrolan Lampu Lalu Lintas

B. Perancangan Hardware

Perancangan *hardware* pada pengontrolan lampu lalu lintas ini menjelaskan tentang suatu program yang nantinya akan diupload ke sebuah PLC yang terhubung secara serial dengan *hardware interface*/sebuah kit yang di dalam nya terdapat sebuah mikrokontroler arduino mega.

1) Rangkaian Penaik Tegangan (5V ke 24V)

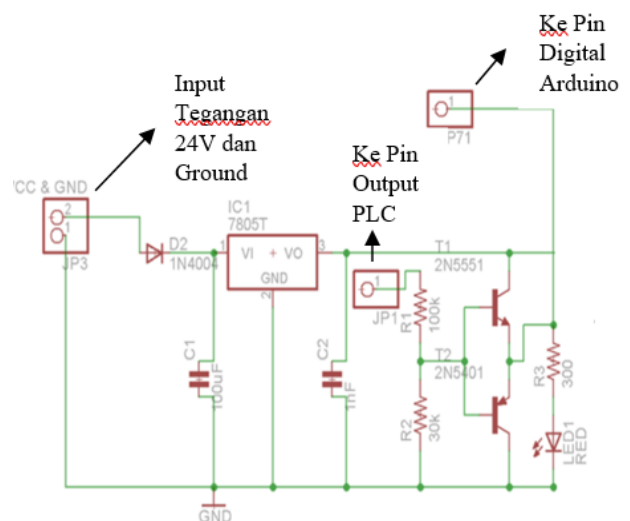
Agar terjadi sebuah koneksi antara Arduino Mega dengan PLC membutuhkan tegangan sebesar 24V [9]. Untuk mendapat tegangan 24V itu diperlukan sebuah rangkaian penaik tegangan, karena Arduino Mega memiliki tegangan output hanya $\pm 5V$. Pada rangkaian ini terdapat beberapa komponen yaitu diode 1N4004 yang berfungsi agar tidak ada arus balik, optocoupler PC817 yang berfungsi sebagai switch, resistor 10k Ω , resistor 100 Ω , resistor 100k Ω , resistor 5k6 Ω berfungsi sebagai tahanan, transistor 2N5551 berfungsi sebagai *switch* (penguat kedua setelah optocoupler) dan LED berwarna merah berfungsi indikator. Rangkaian penaik tegangan dapat dilihat pada **Gambar 2** berikut.



Gambar 2. Rangkaian Penaik Tegangan

2) Rangkaian Penurun Tegangan (24V ke 5V)

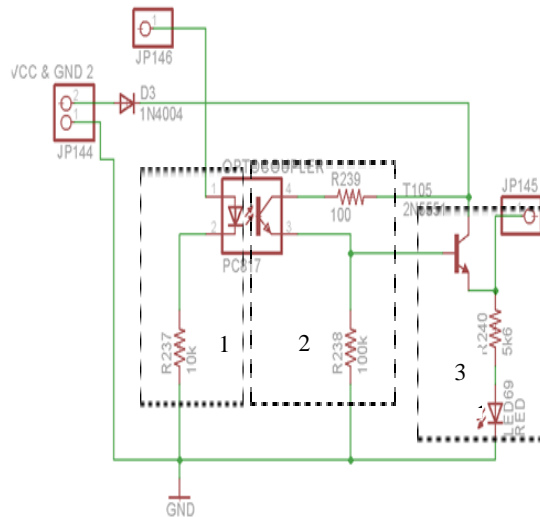
Pada perancangan ini diperlukan sebuah rangkaian pendukung yang dapat menurunkan tegangan, sehingga komunikasi dapat terjadi antara PLC dengan Arduino Mega. Pada rangkaian ini terdapat komponen utama yaitu IC regulator tipe 7805 yang berfungsi menurunkan tegangan dari 24V yang nantinya agar tegangan outputnya stabil $\pm 5V$, sehingga tegangan yang di terima Arduino Mega sesuai [10]. Selain itu ada beberapa komponen lain yang digunakan yaitu dioda 1N4004 yang berfungsi sebagai penyearah, kapasitor 100uF dan kapasitor 1nF berfungsi untuk mengurangi ripple, resistor 100k Ω , resistor 47k Ω , resistor 300 Ω berfungsi sebagai tahanan, transistor 2N5551 & 2N5401 berfungsi sebagai switch LED berwarna merah berfungsi sebagai indikator. Rangkaian penurun tegangan dapat dilihat pada **Gambar 3** berikut.



Gambar 3. Rangkaian Penurun Tegangan

3) Analisis Rangkaian

Pada **Gambar 4** ini terdapat analisis yang harus dilakukan pada rangkaian penaik tegangan sebagai berikut.



Gambar 4. Analisis Rangkaian Penaik Tegangan

Pada bagian 1 pada **Gambar 4** didapatkan suatu persamaan untuk menghitung besaran tahanan yang digunakan sebagai berikut:

$$R = \frac{V}{I} \quad (1)$$

$$R = 25k\Omega$$

Maksimal tahanan atau resistor yang digunakan yaitu 25 k Ω , jika tahanan yang digunakan melebihi batas, maka rangkaian tidak akan berfungsi, karena semakin besar nilai tahanan semakin kecil nilai arus. Pada bagian 2 **Gambar 4** didapatkan persamaan untuk menghitung tegangan yang mengalir sebagai berikut:

$$V = V_{in} \times \left(\frac{R1}{R1 + R2} \right) \quad (2)$$

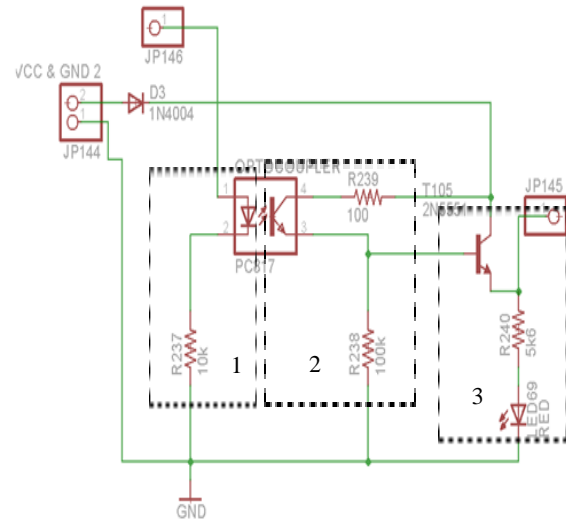
$$V = 23,976V$$

Pada bagian 3 **Gambar 4** didapatkan persamaan untuk menghitung besaran tahanan yang digunakan sebagai berikut:

$$R = \frac{(V_{in} - V_{led})}{I} \quad (3)$$

$$R = 5325\Omega \approx 5600\Omega$$

Pada **Gambar 5** berikut ini terdapat analisis yang harus dilakukan pada rangkaian penurun tegangan sebagai berikut.



Gambar 5. Analisis Rangkaian Penurun Tegangan

Pada bagian 1 **Gambar 5** Penggunaan komponen kapasitor yaitu berfungsi sebagai filter. Fungsi filter untuk mengurangi *ripple* pada rangkaian. Pada bagian 2 **Gambar 5** didapatkan persamaan untuk menghitung tegangan yang mengalir sebagai berikut:

$$V = V_{in} \times \left(\frac{R1}{R1 + R2} \right) \quad (4)$$

$$V = 4,8V$$

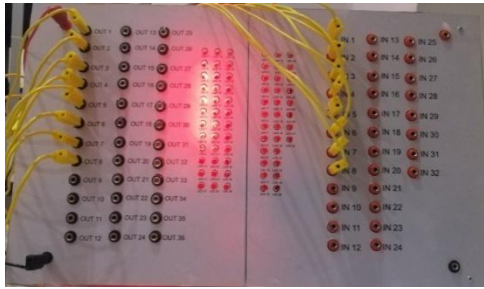
Pada bagian 3 **Gambar 5** didapatkan persamaan untuk menghitung besar tahanan yang digunakan sebagai berikut:

$$R = \frac{(V_{in} - V_{led})}{I} \quad (5)$$

$$R = 460\Omega \approx 300\Omega$$

C. Hasil Perancangan Hardware

Setelah melalui semua tahapan mulai dari membuat rancangan rangkaian penaik tegangan dan membuat rancangan penurun tegangan, berikut hasil implementasi hardware yang mana nanti akan digunakan sebagai pendukung sistem ini dibuat. Hasil rancangan hardware ini terdapat pin out sebanyak 36 pin dan pin in sebanyak 32 pin, serta pin untuk input tegangan 24V dan ground. Hasilnya dapat dilihat pada **Gambar 6**.

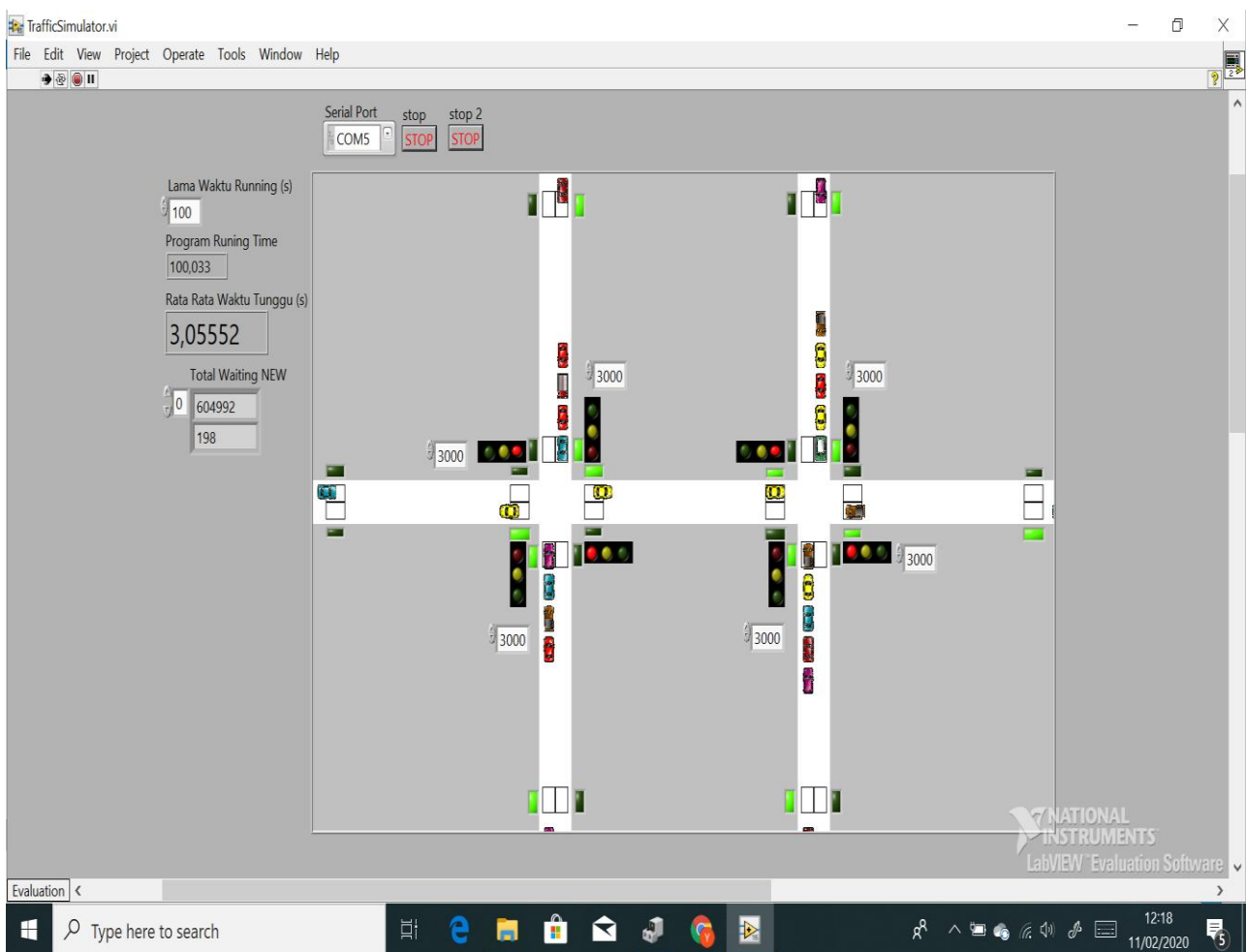


Gambar 6. Implementasi Hardware Interface

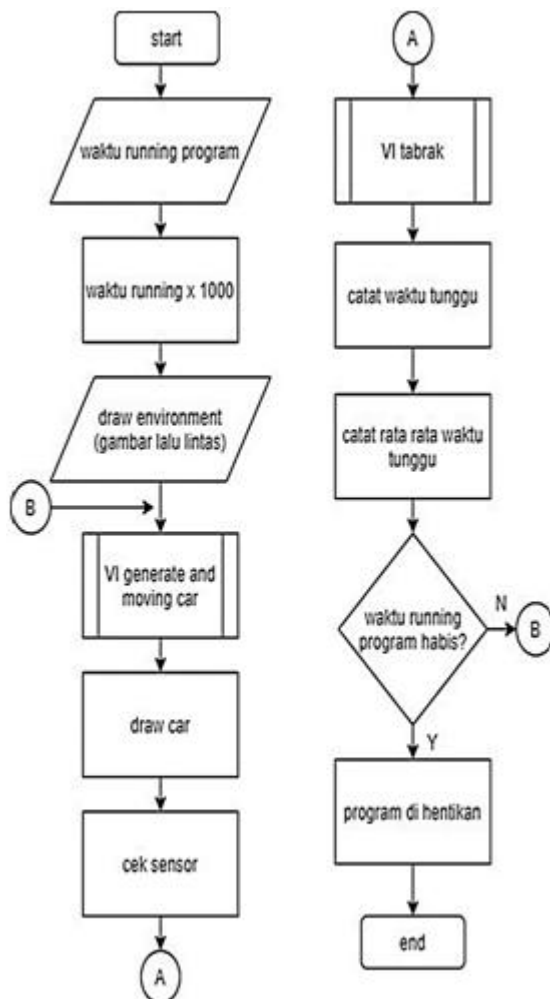
D. Perancangan Simulator

Pembuatan simulator di buat menggunakan aplikasi LabVIEW yang mana di dalamnya terdapat beberapa VI pendukung agar program yang di ingin kan bisa tercapai. Pada bagian dalamnya terdapat beberapa perangkat seperti lampu lalu lintas yang berjumlah 8 buah, dan sensor input output di setiap ruas jalan. Untuk dapat melihat lebih jelas lagi, tampilan simulator dapat dilihat pada **Gambar 7**.

Untuk lebih jelasnya simulator akan di jelaskan dalam sebuah diagram alir (flowchart) pada **Gambar 8 – Gambar 11**. Pada awal program dijalankan pertama akan memasuki wilayah VI moving and generate car yang mana di dalamnya akan mengatur titik stop mobil yang sedang di bangkitkan dan mengatur apakah mobil harus di bangkitkan atau tidak, lalu setelah itu masuk ke VI draw car yang mana di dalamnya mengatur penggambaran dan pemilihan mobil jika ada mobil yang akan di bangkitkan oleh program VI generate and moving lalu setelah itu masuk ke dalam VI sensor untuk mendeteksi kondisi di dalam jalur, apakah di dalam jalur sedang penuh atau tidak, jika di dalam jalur tersebut sedang penuh maka VI generate and moving tidak akan membangkitkan mobil lagi di jalur tersebut dan jika kondisi jalur dalam keadaan kosong maka VI generate and moving akan membangkitkan mobil,

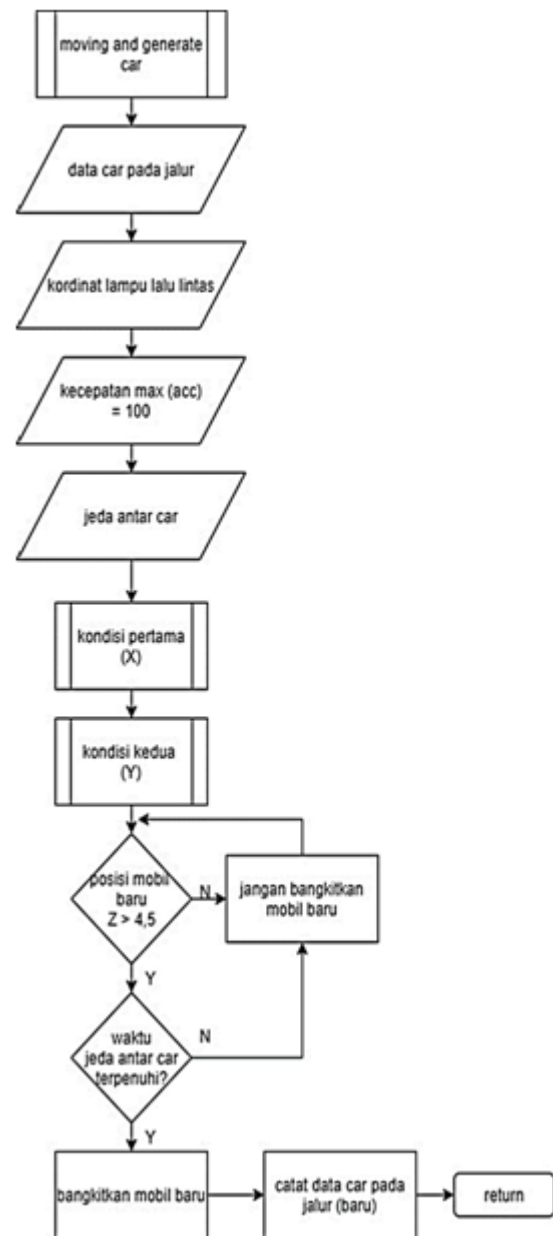


Gambar 7. Tampilan Simulator



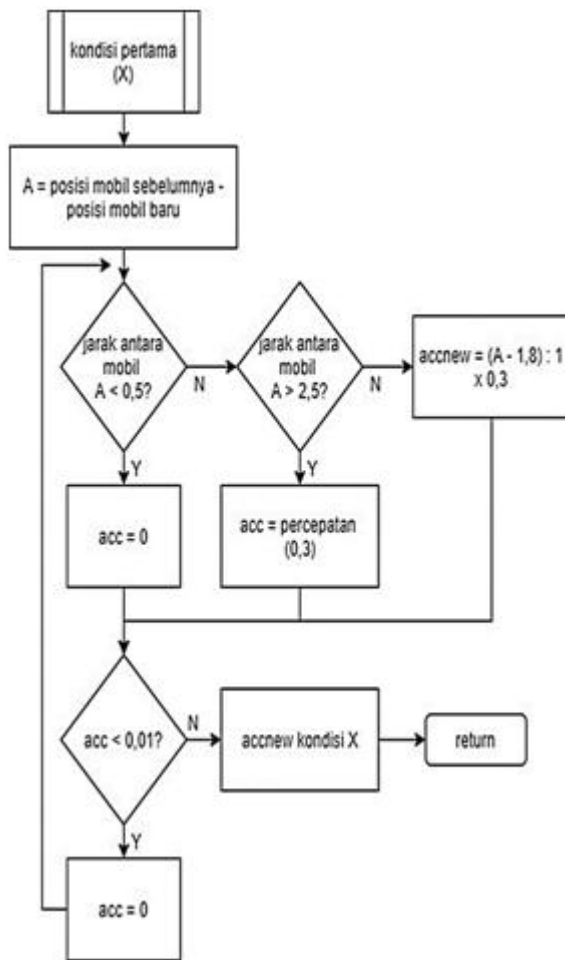
Gambar 8. Flowchart simulator

Lalu setelah mobil di bangkitkan VI generate and moving juga akan mendeteksi apakah di depan mobil yang di bangkitkan ada lampu lalu lintas yang sedang berwarna merah atau tidak, jika di depan mobil yang di bangkitkan terdapat lampu lalu lintas yang sedang berwarna merah maka mobil akan perlahan berhenti dan jika tidak terdapat lampu lalu lintas yang berwarna merah maka mobil yang sedang di bangkitkan akan di beri kecepatan penuh, lalu setelah itu VI generate and moving juga akan mendeteksi apakah terdapat mobil yang sedang berhenti di depan mobil yang sedang di bangkitkan atau tidak, jika di depan terdeteksi ada mobil yang sedang berhenti didepan mobil yang sedang di bangkitkan maka mobil akan berhenti dan jika tidak terdeteksi mobil yang sedang berhenti maka mobil akan di beri kecepatan penuh, lalu setelah itu masuk kedalam VI tabrak yang mana di dalam VI tersebut akan mengatur pendeteksian adanya tabrakan atau tidak di persimpangan jalan yang kordinat tabraknya sudah di tentukan.



Gambar 9. Flowchart generate and moving car

Jika terjadi tabrakan di persimpangan maka program yang sedang di jalankan akan berhenti dan akan adan notifikasi pada simulatur berupa gambar crash dan jika tidak terdeteksi adanya tabrakan maka program akan di lanjut lalu akan masuk ke VI catat wait yang mana VI tersebut berfungsi untuk mencatat lamanya mobil pada suatu jalur berhenti dan menunggu sampai lampu lalu lintas kembali hijau, lalu setelah itu masuk ke VI total wait yang mana fungsi dari VI tersebut mencatat semua total waktu tunggu pada suatu jalur saat mobil berhenti yang nantinya akan di kalkulasikan dengan waktu tunggu jalur lain dan akan di ambil nilai rata – rata tunggu dari semua jalur setelah itu program akan terus berjalan sampai dengan waktu yang telah di tentukan.

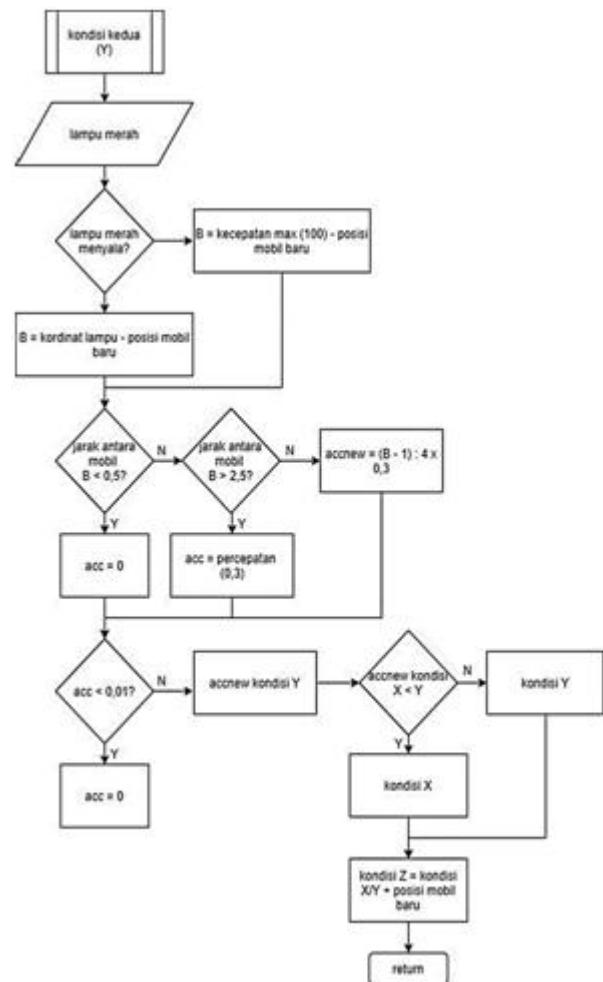


Gambar 10. Flowchart kondisi pertama

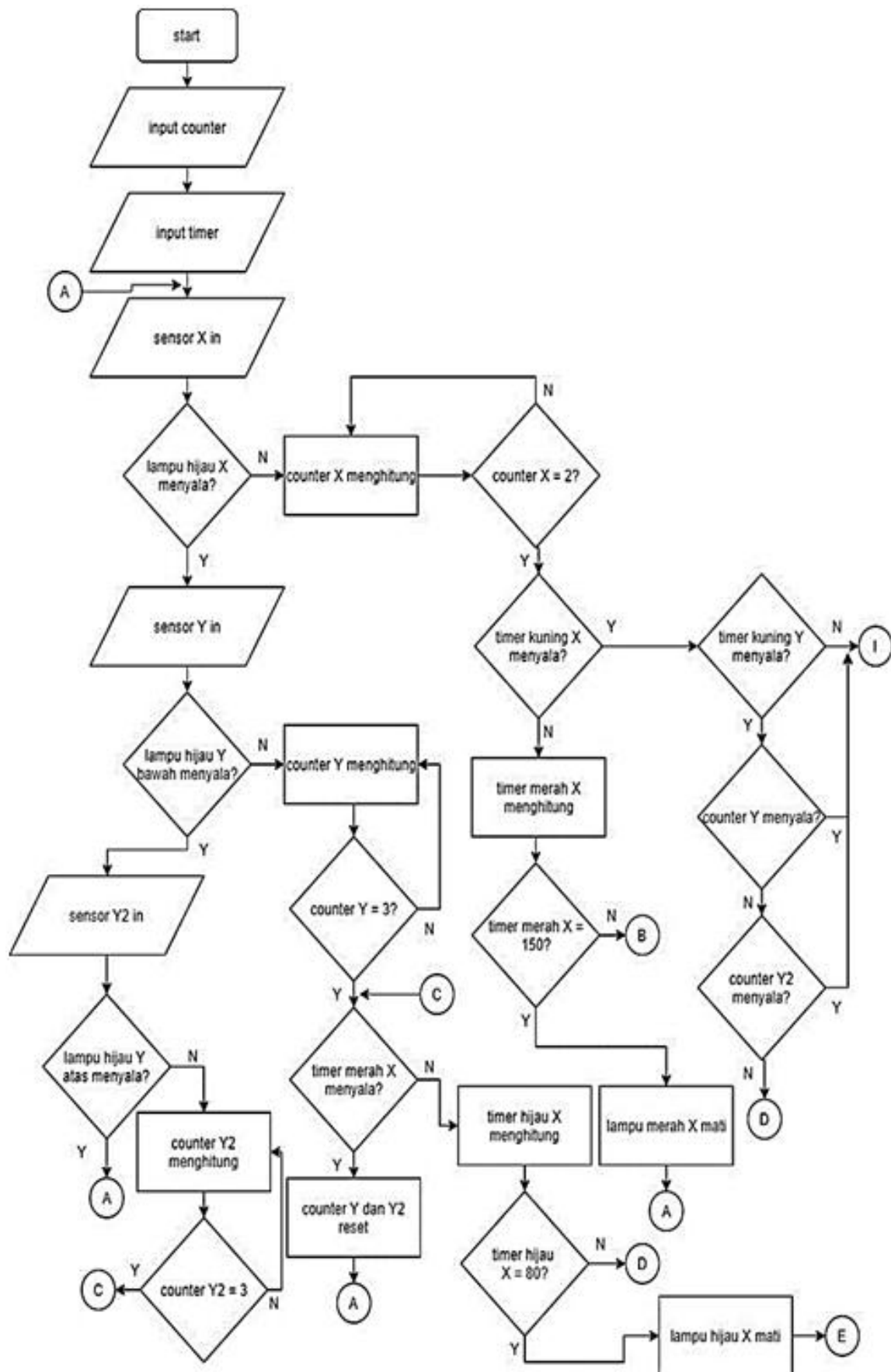
E. Perancangan Pengontrolan Sistem Counter

Perancangan simulator yang telah di lakukan akan di kontrol menggunakan program yang telah di buat menggunakan CX – programmer, yang mana di dalamnya terdapat beberapa intruksi bawaan dari aplikasi tersebut yang di pakai seperti intruksi TIM atau *timer*, intruksi tersebut berfungsi jika diberikan input ON kepadanya, maka setelah selang waktu yang ditentukan, output timer ini akan berubah dari keadaan awal OFF menjadi ON sampai dengan input timer dimatikan, intruksi lain juga yang dipakai dalam pemograman kali ini adalah CNT atau *counter*, intruksi yang jika diberikan input ON setelah beberapa kali seperti yang diinginkan oleh *programmer*, maka *output counter* akan berubah dari keadaan awal OFF menjadi ON sampai dengan kita memberikan input ON pada bagian reset counter.

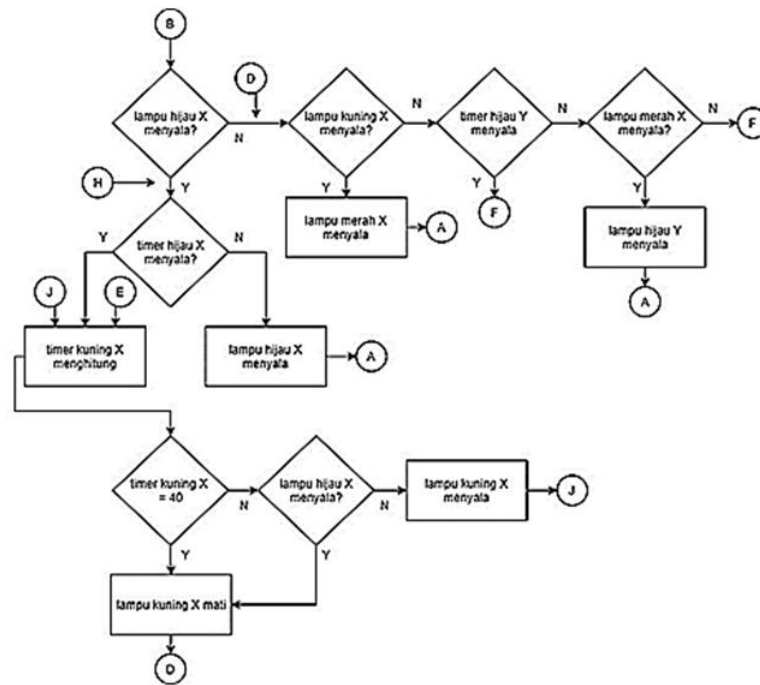
Pada saat program di jalankan kondisi pertama yang akan di ditampilkan adalah lampu merah pada jalur Y akan menyala dan lampu hijau pada jalur X akan menyala, kondisi akan tetap sama sampai counter pada jalur Y akan mendeteksi jalur sudah penuh, suatu jalur di katakan penuh pada saat sensor input pada jalur tersebut terus menyala atau mendeteksi adanya mobil yang melintas atau berhenti pada sensor input tersebut. Untuk lebih jelasnya dapat dilihat pada **Gambar 12 - Gambar 14**.



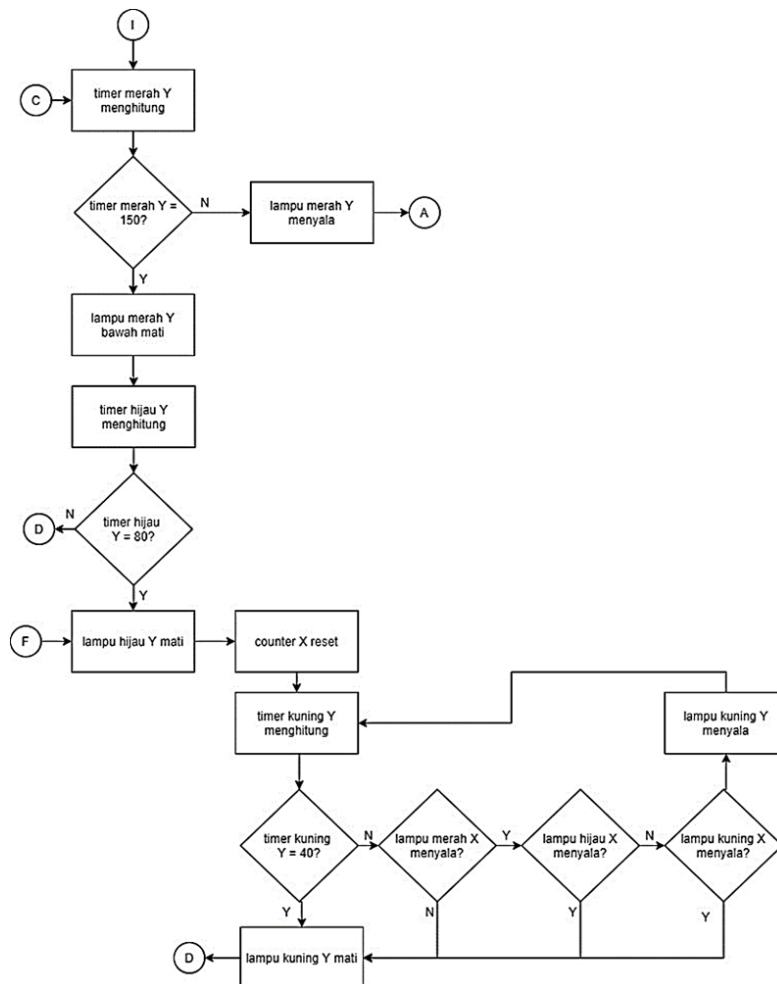
Gambar 11. Flowchart kondisi kedua



Gambar 12 Flowchart plc bagian 1



Gambar 13. Flowchart plc bagian 2



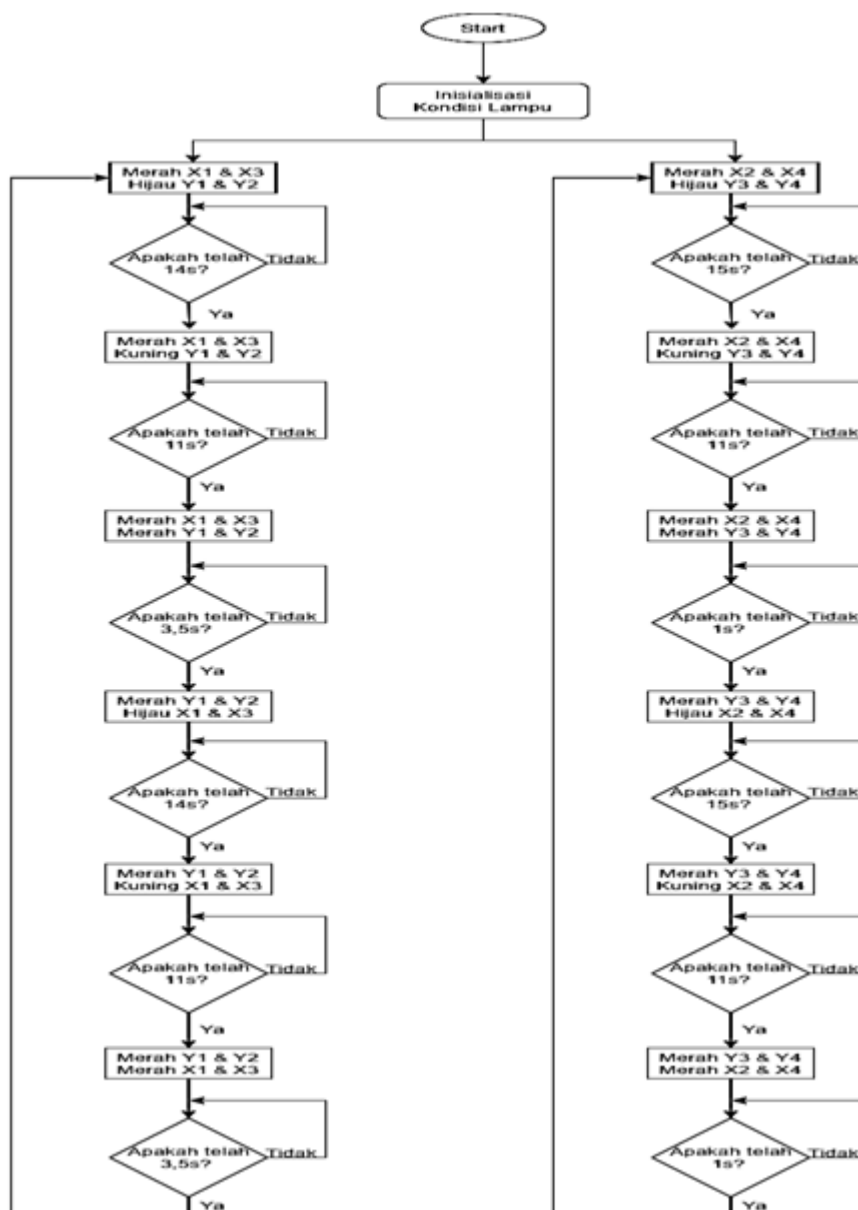
Gambar 14. Flowchart plc bagian 3

F. Perancangan Sistem Pengontrolan *Adaptive Fix Timer*

Dapat dilihat pada **Gambar 15**, bahwa kondisi pertama persimpangan ke - 1 jalur X1 dan X3 merah, Y1 dan Y3 hijau selama 14 s, setelah waktu tercapai akan berubah kondisi, namun bila waktu belum tercapai akan mengulang proses pertama. Kondisi kedua persimpangan ke - 1 jalur X1 dan X3 merah, jalur Y1 dan Y2 kuning lalu hijau selama 11 s, setelah waktu tercapai akan berubah kondisi, namun bila waktu belum tercapai akan mengulang proses kedua. Kondisi ketiga persimpangan ke - 1 X1 dan X3 merah, Y1 dan Y2 selama 3,5 s, setelah waktu tercapai akan berubah kondisi, namun bila waktu belum tercapai akan mengulang proses ketiga. Kondisi keempat persimpangan ke - 1 akan berkebalikan dengan

kondisi pertama persimpangan ke -1, dengan jalur Y1 dan Y2 menjadi merah, X1 dan X3 menjadi hijau, begitupun proses kelima persimpangan ke - 1 akan akan berkebalikan dengan kondisi kedua persimpangan ke - 1, dan kondisi keenam persimpangan ke - 1 akan berkebalikan dengan kondisi ketiga persimpangan ke - 1.

Pada saat kondisi pertama persimpangan 1 berfungsi, kondisi pertama persimpangan ke - 2 akan sama - sama berfungsi tetapi dengan waktu yang berbeda. Pada persimpangan ke - 1 kombinasi pewaktuannya 14 s, 11 s. dan 3,5 s dan Kombinasi pewaktuan pada persimpangan ke - 2 yaitu 15 s, 11 s, dan 1 s, dibedakan pewaktuannya agar mengantisipasi *crash* pada saat dipersimpangan.



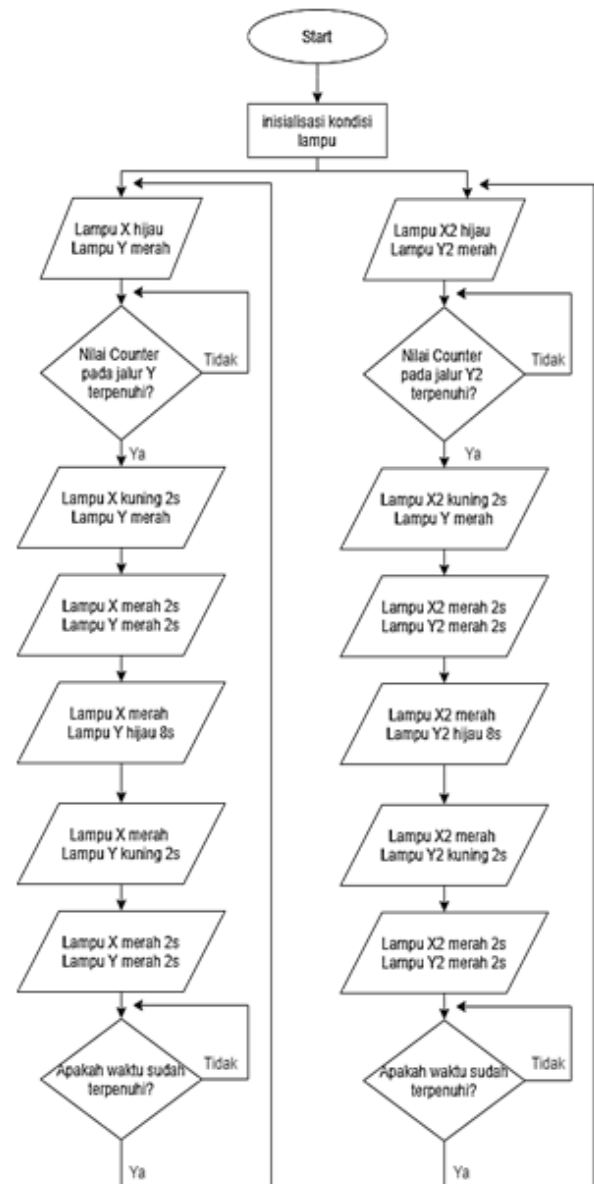
Gambar 15. Flowchart adaptive fix timer

G. Perancangan Sistem Pengontrolan *Semi Intelligent Controller*

Pada **Gambar 16**, kondisi pertama saat pengoprasian lampu lalu lintas berjalan adalah lampu hijau di jalur X akan menyala dengan durasi waktu yang telah ditentukan serta lampu merah di jalur Y pun akan menyala ketika lampu hijau di jalur X menyala. Kemudian jika tidak ada kendaraan yang melintas di jalur Y maka lampu hijau di jalur X dan lampu merah di jalur Y akan terus menyala, akan tetapi jika sensor Y yang ada di jalur Y mendeteksi adanya kendaraan yang melintas di jalur Y, maka akan beralih pada kondisi yang kedua dimana lampu pada jalur X akan berubah dari lampu hijau X menjadi lampu kuning X selama 2 detik dan lampu merah Y akan tetap menyala.

Kondisi kedua ini akan terus berjalan dengan waktu tunggu yang telah ditentukan. Setelah waktu sudah habis maka akan langsung melanjutkan pada kondisi yang ketiga. Dimana kondisi yang ketiga akan menyalakan lampu merah X dan lampu merah Y dengan waktu tunggu selama 2 detik. Setelah kondisi ketiga sudah selesai menunggu selama 2 detik, lampu merah X akan tetap menyala dan lampu hijau Y juga menyala dan ini terjadi pada kondisi keempat. Yang dimana pada kondisi keempat ini akan menyalakan lampu merah X dan lampu hijau Y selama waktu yang ditentukan. Ketika pada kondisi empat ini sudah selesai menunggu selama waktu yang ditentukan maka lampu merah X dan lampu hijau Y akan berubah dari lampu hijau Y menjadi lampu kuning Y dan lampu merah X akan tetap menyala.

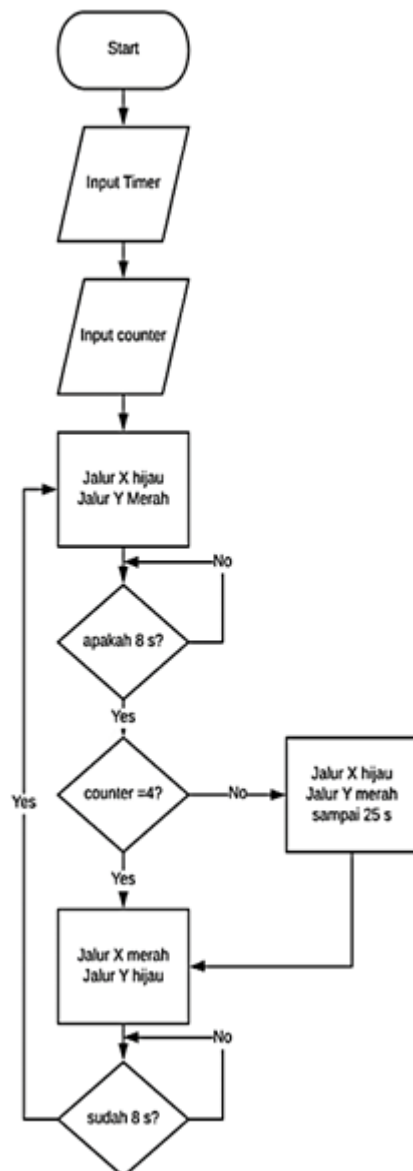
Kondisi kelima ini akan terus berjalan dengan waktu tunggu yang telah ditentukan. Setelah waktu sudah habis maka akan langsung melanjutkan pada kondisi yang keenam. Dimana kondisi yang keenam akan menyalakan lampu merah X dan lampu merah Y dengan waktu tunggu selama 2 detik. Setelah itu, jika kondisi enam sudah selesai maka akan *looping* kembali ke atas atau kembali pada kondisi yang pertama. Begitupun dengan persimpangan yang kedua prosesnya sama seperti persimpangan yang pertama. Oleh karena itu beberapa penjelasan tentang *flowchart* sistem dari program yang dibuat ditunjukkan pada **Gambar 16**.



Gambar 16. Flowchart semi intelligent controller

H. Perancangan Sistem Pengontrolan *Vehicle Actuated*

Dimana saat jalur X hijau dan jalur Y merah, akan menyala selama 8 s, tetapi disaat yang bersamaan akan melihat kondisi kembali counter terpenuhi atau tidak, bila tidak maka akan menunggu selama 25 s. dan bila ya kondisi akan berubah menjadi jalur X merah dan jalur Y hijau. Setelah itu kondisi jalur X merah jalur Y hijau menyala selama 8 s, jika terpenuhi kondisi akan berubah ke kondisi pertama yaitu jalur X hijau dan jalur Y merah, bila tidak maka timer akan menghitung kembali hingga terpenuhi selama 8 s. Dapat dilihat pada **Gambar 17**.



Gambar 17. Flowchart vehicle actuated controller

III. HASIL DAN PEMBAHASAN

Pada pengujian kali ini diambil beberapa contoh kondisi yang nantinya hasil dari pengujian saat ini akan dibandingkan dengan hasil dari pengujian dengan sistem lain yang melakukan pengujian juga dengan pewaktuan dan kondisi yang sama dan akan di tarik kesimpulan dari pembandingan tersebut, sistem yang mana yang paling efektif atau sistem mana yang paling unggul dalam suatu kondisi. Dilakukannya analisis ini bermaksud untuk mencari sistem mana yang total rata – rata waktu tungguanya paling sedikit atau dapat dikatakan paling efektif.

A. Pengujian Sistem Counter

Berikut ini merupakan data yang telah di ambil dari pengujian sistem counter pada simulator dua persimpangan jalan yang berdekatan, pada

pengujian kali ini di ambil 3 kondisi, Berikut ini adalah data hasil pengujian yang sudah di lakukan pada **Table I**.

Tabel I. Data Hasil Pengujian Sistem Counter

No	Kondisi Waktu	Lama Waktu Running (s)	Program Running Time (s)	Rata-Rata waktu Tunngu (s)	Jeda Antara Car X1 (ms)	Jeda Antara Car X2 (ms)	Jeda Antara Car Y (ms)	Banyak Kendaraan Melintas
1	Pagi	300	300,016	2,34589	2000	5000	5000	443
2	Siang	300	300	2,3224	3000	3000	4000	443
3	Sore	300	300,03	3,37335	5000	2000	5000	494
Total Rata – Rata Waktu Tunngu				8,04164	Total banyak kendaraan melintas			1380

Bisa di lihat pada **Tabel I** pada pengujian pertama nilai waktu jeda untuk membangkitkan mobil pada jalur yaitu X1 = 2000, dan yang lainnya di beri nilai 5000, program di jalankan selama 300 detik dan menghasilkan rata – rata waktu tunggu untuk seluruh jalur yaitu 2,34589, pada pengujian kedua nilai jeda untuk membangkitkan mobil pada jalur X2 yaitu 2000 dan jalur sisanya di beri waktu tunggu yang bernilai 5000, program sama di jalankan selama 300 detik dan menghasilkan rata – rata total waktu tunggu sebesar 2,3224, pada percobaan ke tiga nilai jeda waktu pembangkitan mobil pada jalur X1 dan X2 di beri nilai 3000 sedangkan pada jalur Y1, Y2, Y3 dan Y4 di beri nilai 4000, program sama di jalankan selama 300 detik dan menghasilkan waktu rata – rata tunggu sebesar 3,37335, nilai rata – rata total di ambil dari perhitungan antara waktu tunggu dari mobil di semua jalur dan jumlah mobil yang melintas di jalur tersebut dengan persamaan seperti ini. Total waktu tunggu yaitu hasil dari semua waktu tunggu mobil saat berhenti pada suatu jalur dibagi dengan jumlah mobil yaitu jumlah mobil yang di bangkitkan dari semua jalur lalu di bagi lagi dengan 1000 maka di dapatkan hasil yaitu nilai rata – rata waktu tunggu dari semua jalur pada saat program dijalankan. Di bawah ini merupakan perhitungan dari semua percobaan.

B. Pengujian Sistem Adaptive Fix Timer

Pada **Tabel II** merupakan data perbandingan total maksimal rata-rata waktu tunggu dan berdasarkan total kendaraan yang melintas antara sistem *adaptive fix timer* dengan sistem pengontrol yang lainnya.

Tabel II. Data Hasil Pengujian Sistem Adaptive Fix Timer

No	Kondisi Waktu	Lama Waktu Running (s)	Program Running Time (s)	Rata-Rata waktu Tunggu (s)	Jeda Antara Car X1 (ms)	Jeda Antara Car X2 (ms)	Jeda Antara Car Y1,2,3,4 (ms)	Banyak Kendaraan Melintas
1	Pagi	300	300,08	3,04289	2000	5000	5000	429
2	Siang	300	300,031	3,2281	3000	3000	4000	494
3	Sore	300	300,02	4,55516	5000	2000	5000	435
Total Rata – Rata Waktu Tunggu				10,82615	Total banyak kendaraan melintas			1358

Dari 3 data yang disajikan pada **Tabel II** dapat dilihat pada kondisi waktu pagi dengan lama waktu *running* 300 s, jeda antara car X1 2000 ms, jeda antara car X2 5000 ms dan jeda antara car Y1, Y2, Y3, Y4 5000 ms mendapatkan rata – rata waktu tunggu sebesar 3,04289 s dengan Jumlah mobil yang melintas sebanyak 429 kendaraan. kondisi waktu siang dengan lama waktu *running* 300 s, jeda antara car X1 3000 ms, jeda antara car X2 3000 ms dan jeda antara car Y1, Y2, Y3, Y4 4000 ms mendapatkan rata – rata waktu tunggu sebesar 3,2281 s dengan Jumlah mobil yang melintas sebanyak 494 kendaraan. kondisi waktu sore dengan lama waktu *running* 300 s, jeda antara car X1 5000 ms, jeda antara car X2 2000 ms dan jeda antara car Y1, Y2, Y3, Y4 5000 ms mendapatkan rata – rata waktu tunggu sebesar 4,55516 s dengan Jumlah mobil yang melintas sebanyak 435 kendaraan.

C. Pengujian Sistem Vehicle Actuated Controller

Berikut ini akan disajikan 3 data pengujian yang menggunakan pewaktuan yang berbeda karena menggunakan waktu pagi, siang, dan sore. Berikut ini disajikan hasil pengujian *vehicle actuated* sebagai berikut.

Tabel III. Data Hasil Pengujian Sistem Vehicle Actuated Controller

No	Kondisi Waktu	Lama Waktu Running (s)	Program Running Time (s)	Rata-Rata waktu Tunggu (s)	Jeda Antara Car X1 (ms)	Jeda Antara Car X2 (ms)	Jeda Antara Car Y1,2,3,4 (ms)	Banyak Kendaraan Melintas
1	Pagi	300	300,025	3,16749	2000	5000	5000	399
2	Siang	300	300,017	4,99538	3000	3000	4000	469
3	Sore	300	300,025	3,18354	5000	2000	5000	393
Total Rata – Rata Waktu Tunggu				11,34686	Total banyak kendaraan melintas			1261

Pada **Tabel III** data pengujian lampu lalu lintas menggunakan *vehicle actuated controller* dengan melakukan 3 kali percobaan simulasi, dimana hasil rata-rata waktu tunggu kendaraan paling

singkat sebesar 3,16794s pada waktu jeda antara kendaraan X1 2000 X2 5000 dan Y 5000.

D. Pengujian Sistem Semi Intelligent Controller

Berikut ini disajikan hasil pengujian *adaptive fix timer*. Dilihat dari **Tabel IV** data pertama dengan lama waktu *running* kendaraan selama 300 detik, jeda antara car X1 sebesar 2000 ms dan jeda antara car X2 sebesar 5000 ms dan car Y1,Y2,Y3, dan Y4 sebesar 5000 ms maka rata-rata waktu tunggu kendaraan untuk seluruh jalur yaitu 3,47924 s dan banyaknya kendaraan melintas yaitu 443 kendaraan. Percobaan kedua dengan lama waktu *running* kendaraan yang sama yaitu selama 300 detik dengan jeda antara car X1 sebesar 3000 ms dan X2 sebesar 3000 ms dan jeda antara car Y1,Y2,Y3, dan Y4 sebesar 4000 ms maka rata-rata waktu tunggu kendaraan yang dihasilkan adalah 3,04459 s dan banyaknya kendaraan melintas yaitu 494 kendaraan. Sedangkan pada percobaan ketiga dengan lama waktu *running* kendaraan selama 300 detik serta jeda antara car X1 sebesar 5000 ms dan jeda antara car X2 sebesar 2000 ms dan car Y1,Y2,Y3, dan Y4 sebesar 5000 ms yang lainnya sebesar 5000 ms, hasil rata-rata waktu tunggu kendaraan yang dihasilkan adalah sebesar 3,47321 s dan banyaknya kendaraan melintas yaitu 439 kendaraan. Berdasarkan hasil percobaan sistem semi intelligent ini lebih baik digunakan pada kondisi siang hari karena waktu tunggu yang kecil dan jumlah kendaraan yang banyak dibandingkan dengan kondisi pagi dan sore.

Tabel IV. Data Hasil Pengujian Semi Intelligent Controller

No	Kondisi Waktu	Lama Waktu Running (s)	Program Running Time (s)	Rata-Rata waktu Tunggu (s)	Jeda Antara Car X1 (ms)	Jeda Antara Car X2 (ms)	Jeda Antara Car Y (ms)	Banyak Kendaraan Melintas
1	Pagi	300	300,032	3,04289	2000	5000	5000	443
2	Siang	300	300,026	3,2281	3000	3000	4000	494
3	Sore	300	300,028	4,55516	5000	2000	5000	439
Total Rata – Rata Waktu Tunggu				9,99704	Total banyak kendaraan melintas			1376

E. Perbandingan Hasil Data Antara Sistem Adaptive Fix Timer, Counter, Semi Intelligent Controller dan Vehicle Actuated Controller

Pada **Tabel V** data perbedaan total maksimal rata-rata waktu tunggu dan berdasarkan total kendaraan yang melintas antara sistem *adaptive fix timer*, *counter*, *semi intelligent controller* dan *vehicle actuated controller* sebagai berikut.

Tabel V. Perbandingan Hasil Data Antara Sistem *Adaptive Fix Timer*, *Counter*, *semi Intelligent Controller* dan *Vehicle Actuated Controller*

No	Jenis Pengontrol	Rata-Rata Waktu Tunggu (s)			Total Rata-Rata Waktu Tunggu (s)	Banyak Kendaraan Yang Melintas			Total Banyak Kendaraan Melintas
		Pagi	Siang	Sore		Pagi	Siang	Sore	
1	<i>Fix Timer</i>	3,04289	3,2281	4,55516	10,82615	429	494	435	1358
2	<i>Semi Intelligent</i>	3,47924	3,04459	3,47321	9,99704	443	494	439	1376
3	<i>Counter</i>	2,34589	2,3224	3,37335	8,04164	443	443	494	1380
4	<i>Vehicle Actuated</i>	3,16794	4,99538	3,18354	11,34686	399	469	393	1261

Berdasarkan dari total maksimal rata-rata waktu tunggu dan total kendaraan yang melintas yang diambil dari tiga data percobaan pada sistem *adaptive fix timer* yaitu rata – rata waktu tunggu sebesar 10,82615 s dan banyak kendaraan yang melintas sebanyak 1358 kendaraan. Pada sistem *semi intelligent controller* untuk tiga data yaitu rata – rata waktu tunggu sebesar 9,99704 s dan banyak kendaraan yang melintas sebanyak 1376 kendaraan. Pada sistem *counter* untuk tiga data yaitu rata – rata waktu tunggu sebesar 8,04164 s dan banyak kendaraan yang melintas sebanyak 1380 kendaraan. Pada sistem *vehicle actuated controller* untuk tiga data yaitu rata – rata waktu tunggu sebesar 11,34686 s dan banyak kendaraan yang melintas sebanyak 1261 kendaraan.

IV. KESIMPULAN

Berdasarkan hasil perancangan ini dapat disimpulkan bahwa implementasi 4 pengontrol pada PLC untuk pengontrolan lampu lalu lintas dua persimpangan berdekatan telah berhasil dibuat dengan efektif. Perancangan *hardware interface* yang dibuat telah sesuai dengan rangkaian input dan output yang dibutuhkan yaitu sebesar 5V dan 24V. Dimana 5V untuk PLC ke Arduino, dan 24V untuk Arduino ke PLC. Perancangan simulator LabVIEW pun telah berhasil dibuat dan dapat dipergunakan dengan baik. Dari hasil data perbandingan yang dilakukan bahwa tingkat efektivitas pengontrol yang paling baik yaitu pertama pengontrol *counter* dengan total rata – rata waktu tunggu sebesar 8,04164 s, kedua pengontrol *semi intelligent controller* dengan total rata - rata waktu tunggu sebesar 9,99704 s, ketiga pengontrol *adaptive fix timer* dengan total rata – rata waktu tunggu sebesar 10,82615 s, dan keempat pengontrol *vehicle actuated controller* dengan total rata – rata waktu tunggu sebesar 11,34686 s.

DAFTAR PUSTAKA

- [1] Hidayati, Q. (2017). Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 6(2). <https://doi.org/10.22146/jnteti.v6i2.318>
- [2] Jatmika, S., & Andiko, I. (2014). Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler ATMEGA16. *Jurnal Ilmiah Teknologi Informasi Asia*, 8(2), 81–96.
- [3] Tinggi, S., Jakarta, T., & Bidakara, M. (2011). *Buana Suhurudin Putra*, b Romi Satria Wahono, c Rufman Iman Akbar E. 6(2), 77–82.
- [4] Aria M, Riezky F. (2017). Sistem Lalu Lintas Terpadu Embedded Traffic System. *Telekontran*, Vol. 5, No. 2, pp- 83 – 93.
- [5] Fairfield, N., & Urmson, C. (2011). Traffic light mapping and detection. *Proceedings - IEEE International Conference on Robotics and Automation*, 5421–5426. <https://doi.org/10.1109/ICRA.2011.5980164>
- [6] Jamal Al-Nabulsi, Abdelwaddood Mesleh, Adnan Yunis. (2017). Traffic light detection for colorblind individuals, *Applied Electrical Engineering and Computing Technologies (AEECT) 2017 IEEE Jordan Conference on*, pp. 1-6.
- [7] Asaad F. Said, Mehrnaz Kh. Hazrati, Farshad Akhbari. (2016). Real-time detection and classification of traffic light signals, *Applied Imagery Pattern Recognition Workshop (AIPR) IEEE*, pp. 1-5.
- [8] Quan Chen, Zhenwei Shi, Zhengxia Zou. (2014). Robust and real-time traffic light recognition based on hierarchical vision architecture, *Image and Signal Processing (CISP) 7th International Congress on*, pp. 114-119.
- [9] A. El-Dalil, Maha Sharkas, Mohamed Khedr. (2017). Priority level mutualism for emergency vehicle using game theory, *Vehicular Electronics and Safety (ICVES) IEEE International Conference on*, pp. 75-80.
- [10] Harpal Singh, Krishan Kumar, H. K. (2012). Intelligent Traffic Lights Based on RFID. *I-Society*.
- [11] Cao, Z., Jiang, S., Zhang, J., & Guo, H. (2017). A Unified Framework for Vehicle Rerouting and Traffic Light Control to Reduce Traffic Congestion. *IEEE Transactions on Intelligent Transportation Systems*, 18(7), 1958–1973. <https://doi.org/10.1109/TITS.2016.2613997>